UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/809,716	03/24/2004	Kalyan Muthukumar	42P18140	8094

45209 7590 12/22/2008
INTEL/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
VU, TUAN A	
ART UNIT	PAPER NUMBER
2193	
MAIL DATE	DELIVERY MODE
12/22/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/809,716	**Applicant(s)** MUTHUKUMAR ET AL.	
	Examiner TUAN A. VU	**Art Unit** 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11/12/08*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-13,15-26,28-34 and 36-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-13,15-26,28-34 and 36-38* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 11/12/08.

As indicated in Applicant's response, claims 1, 3-5, 7, 9, 13, 15-16, 19, 21, 22, 25-26, 28-33, 37-38 have been amended. Claims 1, 3-13, 15-26, 28-34, 36-38 are pending in the office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.
>
> The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is for the United States Patent And Trademark Office (USPTO) policy on 35 U.S.C. §101. <http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf>

3. Claims 25-26, 28-31 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Specifically, claim 25 recites a apparatus for compiling object code comprising a front end and a code generator, however, all of which elements comprised in said apparatus perceived from the Disclosure as software implemented entities. A front end to receive input code and a back end code generator are components of a software tool, no hardware per se being included in the 'apparatus' claim as to support functional realization of such tool. The claim amounts to listing of mere Functional Descriptive Material (see Annex IV of USC101 Guidelines, pg. 52-54); that is, absent hardware support to materialize functionality of listed code instructions, software per se cannot be construed as a permissible statutory category able to yield a real world result (i.e. concrete, tangible, and useful result provided by data transformation from Practical Application implementing one of the 4 statutory categories) via code execution by machine support. Not only a claimed invention for listing of mere software does not constitute a statutory category of subject matter, but mere provision of software functionality without hardware support cannot be viewed as a Practical Application that would carry out such functionality (yielding tangible, useful, concrete result) into real-world output above the realm of data abstraction.

Dependent claims 26, 28-31 fail to remedy to the above deficiency. Thus, claims 25-26, 28-31 are rejected under 35 U.S.C. 101 because they are directed to non-statutory subject matter.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1, 3, 5, 12-13, 15, 17, 24-26, 28-30, 32-34, 36-38 are rejected under 35 U.S.C. § 102(e), as being anticipated by Kumar et al, USPN: 7,007,271 (hereinafter Kumar)

As per claim 1, Kumar discloses a method comprising:

determining a slack value for each of one or more ready instructions based on resource constraints (e.g. slack value – col. 14, lines 4-31; step 115 – Fig. 1);

selecting one of the ready instructions, based on the slack value (e.g. priority function - col. 14, lines 19-31) of the one ready instruction; and

scheduling the selected ready instruction (e.g. step 105—115, select instruction step 155 – Fig. 1 – Note: select a region of instructions based on analyzing their respective *priority function* reads on ready function being selected);

repeating the method of determining, selecting, and scheduling for each of the one or more ready instructions remaining to be selected (col. 14 lines 41-57 ; step 120 to step 135 Fig. 1) and scheduled until all ready instructions have been scheduled.

Kumar does not explicitly disclose *repeating* for each ready instruction: determining a *new* slack value, selecting a ready instruction based on the *new* value and scheduling the selected instruction. However, Kumar discloses a need to update a priority function, which calculate critical path including resources dependencies, or constraints, such that for each instruction, the referred to constraints supports 'slacks' determination in terms of the possible slots an instruction can be scheduled after considering those constraints. Since critical path updating depends on *priority function*s which determine the proper path to take based on *slacks* and resource constraints, the necessary calculating of slacks necessarily implicated in updating of the *priority function* is strongly suggested. Based on Kumar's repeating the steps of determining (e.g.

updating ... priority functions ... repeating - col. 2 lines 15-20), it would have been obvious for one skill in the art at the time the invention was made to implement the scheduling in Kumar (see step 105, Fig. 1) wherein a iterative process would be applied all selected/ready regions so that for each iteration/stage pertinent to a ready region (containing instruction ready for scheduling) a priority function update (step 145, Fig. 1) for each of said ready instruction would have to be re-computed with obtaining of *new* slack value as set forth above as factors in establishing critical paths by Kumar for the instruction to be scheduled. One would be motivated because when all resources are not available to support scheduling, a heuristic as to recalculate slacks thereby deriving the path being critical to Kumar's scheduling would systematically address all ready instructions for all regions selected for scheduling in light of the necessary update of such *priority function* as endeavored by Kumar under unavailability of related resources information (see Kumar: col 13 line 56 to col. 14 line 3; col. 2 lines 15-20).

As per claim 3, Kumar (in view of the rationale in claim 1) discloses determining a slack value (priority function - col. 14, lines 15-31; Fig. 1) for each of the one or more ready instructions further comprises determining the slack value for the instruction based on resource constraints and dependence height (e.g. *total number resources, maximum latency dependencies, larger of the dependencies or resource constraints, longest delay … resource, latency*– col. 12, line 62 to col. 14, line 5).

As per claim 5, Kumar (in view of claim 1) discloses selecting one of the ready instructions further comprises selecting the ready instruction having a lowest new slack value (priority function 115, Fig. 1; *earliest scheduling time ... least number of slacks* – col. 14, lines 4-25).

As per claim 12, Kumar discloses wherein: the resource constraints include the maximum number of instructions of a particular instruction type (e.g. MAX Latency 3 col. 13, lines 3-5; *longest path ... for scheduling, longest delay, longest scheduling time, number of possible slots an instruction can be scheduled* - col. 13 line 64 to col. 14, line 9) that can be scheduled during a given cycle for a selected target processor.

As per claim 13, Kumar discloses an article comprising: a computer readable medium having a plurality of machine accessible instructions stored thereon, which when executed by a computer, cause the computer to perform the following method:

determining a slack value for each of one or more ready instructions in a scheduling region based on resource constraints (refer to claim 1);

selecting one of the ready instructions, based on the slack value (refer to claim 1) of the one ready instruction; and

scheduling the selected ready instruction (refer to claim 1); and repeating the method of determining, selecting, and scheduling for each of the one or more ready instructions remaining to be selected (col. 14 lines 41-57 ; step 120 to step 135 Fig. 1) and

repeating determining, selecting and scheduling until all ready instructions have been scheduled.

all of which operations having been addressed in claim 1.

Kumar does not disclose *repeating* for each ready instruction: determining a *new* slack value, selecting a ready instruction based on the *new* value and scheduling the selected instruction. But this limitation has been addressed in claim 1.

As per claim 15, refer to claim 3.

As per claim 17, Kumar discloses that the selected ready instruction is having highest scheduling priority: since Kumar sets forth an instruction for scheduling and updating this scheduling (col. 13-14) by a heuristics operating on the result of a *priority function* (e.g. step 115 - Fig. 1), Kumar also discloses that priority is given to said instruction by virtue of the priority function computing.

As per claim 24, refer to claim 12.

As per claim 25, Kumar discloses apparatus for compiling a high-level programming language into object code, comprising:

a front end to receive a source code; and a code generator coupled with the front end to receive the source code and compile the received source code into an object code;

wherein the code generator includes one or more resource-aware schedulers to:

determine a slack value …; selecting one of the ready instructions …; scheduling the selected ready … ; and repeating … determining, selecting … based on slack value of one ready instruction, and scheduling … and

repeat ... have been scheduled; all of which having been addressed in claim 1.

Kumar does not disclose *repeating* for each ready instruction: determining a *new* slack value, selecting a ready instruction based on the *new* value and scheduling the selected instruction. But this limitation has been addressed in claim 1.

As per claim 26, Kumar, in view of the rationale of claim 4 or 16, discloses wherein for each of the one or more ready instructions, one or more resource-aware schedulers:

determine a first scheduling deadline for an instruction in a scheduling region, *taking dependence considerations into account*;

determine a second scheduling deadline for the instruction, *taking resource constraints into account*; and said one or more resource-aware schedulers is further to select between the first and second scheduling deadlines to choose a scheduling priority (i.e. a selected slack value) for the instruction.

As per claim 28, Kumar discloses wherein said resource-aware scheduler is further to select the instruction for scheduling based on its scheduling priority (re claim 17).

As per claim 29, Kumar discloses the maximum number of *instructions (e.g. number of all possible ... an instruction can be scheduled* – col. 14, lines 3-11) that can be scheduled per cycle.

As per claim 30, refer to claim 12.

As per claim 32, Kumar discloses a system comprising: a processor to execute each of one or more ready instructions; and a memory coupled to the processor to store each one or more ready instructions (e.g. *compilers ...code generation* - col. 1 lines 8-58); wherein the instructions include a resource-aware scheduler to

determine, based on ...; to select ... ; to schedule ... and to repeat ... (refer to the corresponding mapping in claim 25)

As per claim 33, Kumar discloses wherein: the memory system includes a DRAM (e.g. memory load instruction – col. 21, lines 8-9).

As per claim 34, refer to rationale as set forth in claim 26.

As per claims 36-38, refer to claims 28-30, respectively.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) a patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 4, 9-11, 16, 21-23, 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kumar et al, USPN: 7,007,271.

As per claim 4, Kumar determining a slack value for each of the one or more instructions further comprises: determining a dependence basis based on a dependence height for the instruction (e.g. graph D … 7 edges – col 4, ines 41-45; Table 2, col. 4; Table 4-5, col. 5 – Note: spanning a incremental graph to color regions based on register allocations reads on total height of such spanned regions being colored); determining a resource basis based on resource constraints for the instruction (refer to claim 10); selecting between the resource basis and the dependence basis to choose a slack value that indicates a least number of cycles (e.g. *longest path … for scheduling, longest delay, longest scheduling time, number of possible slots an instruction can be scheduled* - col. 13 line 64 to col. 14, line 9); and determining the slack value based on the selected dependence/resource basis value (e.g. col. 13, line 63 to col. 15, line 25).

But Kumar does not explicitly teach that resource basis as number of cycles between resource deadline and dependency deadline is *deadline value*; nor does Kumar specify that such dependence (i.e. tree region span) or resource basis (i.e. register analysis and region coloring) is respectively called *dependence deadline* and *resource deadline*. However, slack calculation by Kumar entails either a maximum of latency, path size or slots to be scheduled per region spanned by a class of instruction (col. 14 line 8-31; re claim 10-11) based on resources or spanning of sub-region in term of size of edge dependencies (see claim 11), hence the concept of end limit is integral to those tree size basis or register dependency basis whereby computing a number of cycles between the space of the tree payload and the register resources available therefor would be indicative of this measure of *deadline value* taught as Kumar's resource basis represented by number of cycles. It would have been obvious for one skill in the art at the time the invention was made to implement a deadline value for each of said tree spanning analysis and register-based coloring basis by Kumar so that a slack value can be based on selecting between either a *dependence deadline* (sub-tree size) or a *resource deadline* (register available resources) in view of Kumar's optimizing a value based on path spanning and delay analysis as set forth in the priority function heuristics as set forth for this slack value deriving (e.g. Fig. 1; col. 14 lines 4-31)

As per claims 9-10, Kumar discloses for each of the one or more ready instructions, a slack value determining such that it comprises: a minimum number of cycles needed to schedule the instructions of a scheduling region, taking resource constraints into account (*earliest scheduling time ... least number of slacks* – col. 14, lines 4-25); determining dependence deadline and resource deadline based on dependence height and minimum number of cycles (by virtue of the rationale set forth in claim 4); determining dependence length and resource length (refer to claim 4: sub-tree height and register resources therefor).

Kumar does not explicitly disclose assigning dependence length as minimum number of cycles when this dependence length is greater then that of the resource length; and assigning resource length as minimum number of cycles when resource length is greater than dependence length. However, Kumar discloses priority function based on incremental building of a state machine and this reads on being **dependent of graph length** per region (e.g. *minimum number of cycles*, dependence … constraints – col. 13, lines 56-63; FSA automata – col. 11) and register coloring function based on incremental building of a state machine and this reads on being dependent of **resource length** of a region (minimum number of cycles … *resources constraints* - col.13, lines 56-63; graph coloring – Table 15); hence the selection of resource constraints or size of graph region being a constraints suggests a basis by which to evaluate a priority function with choosing slack value represented by said minimum number of cycles. It would have been obvious for one skill in the art at the time the invention was made to implement Kumar's selecting of this slack value via assigning minimum number of cycles based on dependence length (or resource length) according to whether dependence length (or resource length) is respectively greater to one another, because using the greatest (dependence or resource) length as basis as to assign this minimum cycles would better support priority functions as endeavored by Kumar, via graph coloring, register update and Finite state study, which requires estimation of much greatest cost down the remaining path of the regions of the tree under scheduling in Kumar (see Fig. 1).

As per claim 11, Kumar discloses calculating the dependence length of the scheduling region based on the total height (e.g. graph D … 7 edges – col 4, ines 41-45; Table 2, col. 4; Table 4-5, col. 5 – Note: spanning a incremental graph to color regions based on register allocations reads on total height of such spanned regions being colored) of a dependence graph of the scheduling region; and calculating the resource length of the scheduling region based on the maximum number of cycles needed (e.g. R total number resources, maximum latency instruction class – col. 11, lines 45-53; col. 12, lines 62-64; col. 13, lines 3-6) to schedule the instructions of the scheduling region for a machine resource.

As per claim 16, refer to claim 4.

As per claims 21-23, refer to claims 9-11, respectively.

As per claim 31, Kumar does not explicitly disclose wherein the one or more resource-aware schedulers further to schedule the instructions such that instructions of a particular instruction type **are distributed** evenly among two or more resources. But based on the hardware resources analysis and the dependencies of tree spanning and finite state recording by Kumar (see incremental interference graph, graph coloring, finite state automata, Finite state table generation col. 6-13;) leading to *priority function* computing to update the number of registers (see steps 175, 180 – Fig. 1) for a best slack value (col. 14, lines 5-35), it would have been obvious for one skill in the art at the time the invention was made to implement the above register update so that based on the slack value approach, Kumar reschedules a type of instruction taking into consideration register hardware and pertinent architecture (see step 155, Fig. 1) so that it is evenly distributed among the available registers and implementing these schedule algorithms according to hardware-based architecture, because as a result of the role played in Kumar's front end creation of incremental dependence graph and finite state building coupled to graph coloring as set forth above prior to runtime, such hardware and architectural consideration with resulting redistribution of instructions addresses all the limitations imposed by the architecture, its breadth of instruction availability and its underlying register or instruction type restraints.

8. Claims 6-8, 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kumar, further in view of Cooper et al, 'An Experimental Evaluation of List Scheduling', , Rice University, , pp. 1-15 (hereinafter Cooper) – September 1998, date as of TR98-326 (refer to Technical Reports pg. 12, bottom, in provided Devika Subramanian, porfolio)

As per claims 6, 8, Kumar teaches heuristics to build into a state graph per advance of cycle and evaluate metrics per graph node upon each edge transition (see col. 11; Fig. 1) but does not explicitly disclose generating an entry in a ready list for each of the one or more ready instructions; and removing the entry for the selected ready instruction from the ready list; nor does Kumar disclose advancing a virtual clock to a subsequent clock cycle when there are no ready instructions in the ready list; and adding an entry to the ready for list for any non-ready instruction that becomes ready in the subsequent clock cycle.

Analogous to a graph heuristics building by Kumar, Cooper also teaches computing metrics related to minimum schedule length in the context of evaluation critical path (sec 5, pg. 7-9) similar to Kumar's priority function (Kumar: heuristics … critical path – col. 13-14); and further provides computation of priority function (sec 3-4, pg. 3-4 – which is analogous to Kumar's Fig. 1) related to graph depth and cycle latency heuristics with a algorithm, and according to which algorithm Cooper teaches a per-time-cycle (clock cycle) *ready list* of DPG nodes and remove nodes therefrom (see *ready-list, time cycle* - Algorithm pg. 4) in the course of calculating their respective priorities (Input, Output Data Precedence Graph – pg. 4, top). It would have been obvious for one skill in the art at the time the invention was made to implement the heuristics by Kumar so that each instruction identified from a dependence graph per cycle analysis (Kumar: Fig. 1) would be inputted into a ready list as by Cooper for calculating constraints metrics therefor per cycle clock advance, including removing the 'ready' instruction from said list once this evaluation is done; because this is how Kumar intends to find the optimum value for each and every instructions pertinent to a region of dependency tree, as evidenced by Cooper, such that a 'ready' list is created and depleted in function of the traversal of node in said tree by the algorithm.

As per claim 7, Kumar does not explicitly disclose adding to an uncover list any non-ready instructions uncovered by the scheduling of the selected ready instruction wherein the non-ready instructions are dependent on the selected ready instruction. But Cooper teaches a heuristics by which a covering of a instruction from a ready-list a instruction is identified (or uncovered, i.e. as non-ready) as being entered in another list from an evaluation covered under some predicate (Cooper: *if (a functional unit ... and add to the inflight-list* – Algorithm pg. 4). Based on the underlying rationale that selection of one instruction -- as in Kumar's scheduling scheme -- precludes another from being selected (non-ready instructions not selected being dependent of selection of the ready ones) one would be motivated to implement Cooper's uncover list to Kumar's heuristics to better segregate ready list from non-ready list as set forth above from a *uncovered* list, and thereby keeping track of the amount of instructions per clock already covered from executing Kumar's priority calculation algorithm, based on the rationale as set forth in claim 6.

As per claims 18, 19, 20, refer to the rationale of claims 6, 7, 8, respectively.

Response to Arguments

9. Applicant's arguments filed 11/12/08 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 101 Rejection:

(A) The amended claim includes no hardware within the apparatus claim; hence is not sufficient to overcome the non-statutory deficiency.

35 USC § 102 Rejection:

(B) The current rejection has been necessitated by the change to the claim language; hence the argument (Appl. Rmrks pg. 15-16) regarding 'repeating … based on new slack value' is become moot with respect to the previous Office Action.

(C) Applicants have submitted that Kumar disclosing of priority functions per selected region is not same as 'repeating … for determining, selecting and scheduling' with 're-computing priority function (slack value) based on the current resource constraints' (Appl. Rmrks pg. 17). The above allegation does not flow out from any exact language or derived teaching gathered from the previous set of claimed subject matter being prosecuted in the last Office Action; hence would be deemed not conforming with a proper CFR 111b requirement.

(D) Applicants have submitted that updating of priority function in Kumar is not taught as completed and updated, but rather effectuated based on constraints to allow further scheduling of the instruction (Appl. Rmrks pg. 18, top). The disagreement to the Office Action is not commensurate with 1) the state of the subject matter being submitted previously, and 2) with the cited portions and current grounds of rejection. It appears as though Applicants seek patentability of a new set of claimed subject matter while rebutting a grounds of rejection and cited portions therefor that was only in place to address a older subject matter; which cannot be deemed a proper prima facie case of factual rebuttal.

USC § 103 Rejection:

(E) Applicants have submitted that Kumar and Cooper in combination cannot fulfill prima facie (Appl. Rmrks pg. 20-21); but the currently submitted subject matter is such that new grounds of rejection has been effectuated by necessity. Withdrawal of any 103 Rejection would be considered when, with respect to the actual and present claimed invention, **proper and clear factual rebuttal** (i.e. compliance in terms of pointing out how a claim language – not a outdated language - is matched with a cited portions or not, for example in case of a 102/103 rationale) to the current Office Action **is provided**.

The claims as amended will stand rejected as set forth in the Office Action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

December 18, 2008